

Haorui Zhang

1/26/2019

### HW 3

1.

1. X -> \$S0, Y -> \$S1  
addi \$S2, \$S1, 5                   #s2 = y+5  
slt \$t1, \$S0, \$S2                #If x < s2, t1 = 1, else t1 = 0  
beq \$t1, \$ZERO, else            #if x > y+5, go to else  
    sub \$S1, \$S0, \$S1            #y = x - y  
    j EXIT                        #go to exit  
else:  
    add \$S0, \$S0, \$S1            #x = x + y  
EXIT #Quit
2. X -> \$S0, Y -> \$S1  
Loop:  
    addi \$S2, \$S1, 5            #s2 = y+5  
    slt \$t1, \$S0, \$S2            #if x < y+5, t1 = 1, else t1 = 0  
    beq \$t1, \$ZERO, else        #if x > y+5, go to else block  
        EXIT                    #if x < y+5, quit  
else:  
    addi \$S0, \$S0, -1 #x--  
    addi \$S0, \$S0, -1 #x--  
    addi \$S1, \$S1, 1 #y++  
    J Loop                       #back to loop
3. X -> \$S0, Y -> \$S1  
addi \$S0, \$ZERO, 1            #set x = 1  
Loop:  
    sgt \$t1, \$S0, \$S1            #if x > y, t1 = 1, else, t1 = 0  
    beq \$t1, \$zero, else        #If x < y, go to else block  
        Exit                    #If x > Y, quit  
else:  
    addi \$S0, \$S0, 5 #x = x + 5  
    addi \$S1, \$S1, 1 #y++  
    addi \$S0, \$S0, 1 #x++  
    j Loop

C:\Users\made\Desktop\CS260 HW\HW3\HW3(2).asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

**Edit Execute**

**Text Segment**

Block	Address	Code	Basic	Source
0x00400000	0x3e011001	lui	\$1, 4097	8: la \$a0, myArray #a0 -> address of myArray
0x00400004	0x24300000	ori	\$16, \$1, 0	
0x00400008	0x24080000	addiu	\$9, \$9, 0	9: li \$t0, 0 #t0 -> count and tracking index
0x0040000c	0x29090006	slti	\$9, \$9, 6	11: slti \$t1, \$t0, 6 #if t0 < 5, t1=1, else t1=0
0x00400010	0x11200006	beqz	\$t1, \$t0, 6	12: beqz \$t1, else #if index > 4, go to else block
0x00400014	0x24020005	addiu	\$2, \$0, 5	13: li \$v0, 5 #ask for the int input from keyboard
0x00400018	0x0000000c	syscall		14: syscall #call 08
0x0040001c	0x24020000	sw	\$v0, 0(\$a0)	15: sw \$v0, 0(\$a0) #store int to \$a0
0x00400020	0x22100004	addi	\$16, \$16, 4	17: addi \$a0, \$a0, 4 #load next element in array, \$t0 -> index, \$a0 = \$a0 + 4
0x00400024	0x21080001	addi	\$9, \$9, 1	18: addi \$t0, \$t0, 1 #keep track of array element \$t0++
0x00400028	0x08100003	j	0x0040000c	19: j loop #Back to loop
0x0040002c	0x0810000c	j	0x00400030	22: j Compare #Go to compare

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	2	-10	3	-9	-7	23	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

**Registers** Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	10
\$v1	3	0
\$a0	4	3
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	268501016
\$t1	9	0
\$t2	10	268501012
\$t3	11	268501000
\$t4	12	5
\$t5	13	23
\$t6	14	3
\$t7	15	0
\$a0	16	268501016
\$a1	17	0
\$a2	18	0
\$a3	19	0
\$a4	20	0
\$a5	21	0
\$a6	22	0
\$a7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194524
hi		0
lo		0

**Mars Messages** Run I/O

2  
-10  
3  
-9  
-7  
23  
0

Clear

- 2.
3. Since addi instruction works with 16 bits number, the smallest number for this instruction is: - 32768.